SYSTEM AND METHOD FOR COMPILING IMAGES FROM A DATABASE AND COMPARING THE COMPILED IMAGES WITH KNOWN IMAGES

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates generally to computer networks. More particularly, this invention provides for a system and method for searching and compiling from a database, such as the Worldwide Internet, images that have a specified visual content, and for determining if any of the compiled images are substantially similar to one or more known images.

2. Description of the Background Art

In its embryonic stage the Worldwide Internet provided a research-oriented environment where users and hosts were interested in a free and open exchange of information, and where users and hosts mutually trusted one another. However, the Internet has grown dramatically, currently interconnecting over 100,000 computer networks and several million users. Because of its size and openness, the Internet has become a target of trademark and service mark infringement or misuse. Virtually every trademark or service mark is available for unauthorized use on the Internet. Before connecting, companies balance the rewards of an Internet connection against risks of infringement of trademarks and servicemarks.

An entity's brands, trademarks, or servicemarks may be its most valuable asset. This is especially true with global intellectual property such as brands, trademarks, or servicemarks where integrity of the brand, trademark or servicemark is vital in new markets. Unfortunately, piracy of such intellectual property in many of these markets already costs leading corporations billions of dollars in lost sales annually, including new forms of piracy on the Worldwide Internet. Brand images (or look-alike marks) can be surreptitiously posted on web pages for selling fraudulent or unauthorized goods to a global market. If the presence of any brand, trademark or servicemark on the Internet becomes compromised, the result can be dilution of such any brand, trademark or servicemark, and ultimate loss of market share.

In the Worldwide Internet the number of web sites and the number of images increases daily by millions. Right now, there are expected to be more than 500 million images in the

Internet. While searching for regular text in the Internet is known (e. g., commercial text search engines like Yahoo, Altavista, Lycos, etc.), searching solely for images is much more difficult. Presently, searching for images in the Internet is possible only by looking at an image name, e.g., "Clinton.gif," or by looking at the text grouped around an image in a website (e.g., commercial sites like "richmedia.lycos.com," Altavista image finder, etc.). It is believed that there is presently no feasible system to efficiently search for images in the Internet by specifying their visual content, because no computer system or computer method is presently available to detect the specified visual content of an image from all of the millions of images provided in the Internet.

Therefore, what is needed and what has been invented is a system and method for searching and compiling from a database, such as the Worldwide Internet, images that have a specified visual content, and for determining if any of the compiled images are substantially similar to one or more known images. What has been more specifically invented is a high-precision, automated visual detection service to protect global trademarks, servicemarks, and brands from infringement, dilution, or tarnishment by look-alike or imposter marks and brands on the Internet. The visual detection technology provided by the present invention finds a brand, trademark, or servicemark on Internet web pages, and also finds designs, symbols, shapes, and signs that closely resemble the brand, trademark or servicemark. The present invention also identifies logos within a larger picture and text within images.

SUMMARY OF THE INVENTION

The present invention broadly provides a system and method for searching and discovering from a database (e.g., the Worldwide Internet) an object (e.g. a logo, a trademark, etc.) which is confusingly similar with a known object. Broadly, an object crawler sweeps websites of the Internet by automatically following hyperlinks contained in the websites. On each website the object crawler identifies all objects and duplicates them by downloading them on servers of a temporary storage system. Broadly further, after the object are downloaded by the object crawler and stored on the servers of the temporary storage system, the visual content of the objects may be analyzed, such as by hundreds of parallel computers analyzing object content. This may be done in a massive parallel manner with hundreds of computers (e.g., three hundred computers or more). Each computer object operates an object analysis software component which processes one or more input objects and produces as output descriptive information in terms of text and numbers about what content is in the object(s). For each object the following information may be produced and stored: object size; "fingerprint" for efficient identification of substantial similar objects; all text contained in the object(s); "fingerprint" of each face contained in the object(s); information about the logos/trademarks contained in the object(s); and information about things and images contained in the object(s).

In one embodiment of the present invention, a graphical user interface is provided where the user may enter search criteria for the object to be searched. The search criteria to be entered in the graphical user interface may include one or more of the following search criteria: (i) one or more text strings that may be contained in the object including any image; (ii) one or more logo, trademarks or servicemarks selected from a list of predefined logos, trademarks or servicemarks that may be contained in the object including any image; (iii) one or more things or physical features or shapes selected from a list of predefined logos, trademarks or servicemarks that may be contained in the object including any image; (iv) one or more faces of facial templates that may be contained in the object including any image; and (v) one or more images that look substantially similar.

In another embodiment of the present invention, a system and method is provided for searching for an entity's logos, trademarks or servicemarks in objects and images in the Worldwide Internet. A known logo and/or trademark and/or servicemark is provided and is entered into the system of the present inventions; and the content of each object in the internet is compared with the known logo and/or trademark and/or servicemark to determine if there is any confusing similarity. If a confusingly similar logo and/or trademark and/or servicemark appears in the internet object, a reference to the internet object is stored as search results. After scrutinizing objects in the internet, the user may access the search results.

In yet another embodiment of the present invention, a system and method is provided for searching for faces of people or animals that are substantial identical to a known face. The system and method of embodiments of the present invention accept as input an object (e.g., a scanned photograph) that contains at least one face. Subsequently, the input face is compared with all faces in the internet objects (including images) using already computed face "fingerprints" available in storage. The result of the comparison is output in the form of a list of substantial identical objects (including images) that contain a face that is similar or substantially identical to the input face.

Embodiments of the present invention more specifically provide a method for discovering from a database (e.g., the Worldwide Internet) an object which is confusingly similar with a known object comprising: (a) searching (e.g. searching with a web crawler by following hyperlinks contained in web site elements) a database for objects; (b) providing a known object; and (c) determining if any object from the database is confusingly similar with the known object. The method preferably additionally comprises duplicating the objects from the database to produce duplicated objects; storing the duplicated objects to produce stored duplicated objects; and determining if any stored duplicated object is confusingly similar with the known object. The method further preferably additionally comprises determining the degree of similarity of any stored duplicated object with the known object. The objects may be selected from the group consisting of graphic images, videos, audio sounds and mixtures thereof. Each of the objects may be an intellectual property selected from the group consisting of logos, trademarks, service marks, and mixtures thereof. Determining if any object is confusingly similar with the known object further preferably comprises determining if all of the necessary metadata is available for any of the stored duplicated objects; and if not, the necessary metadata is developed for the

stored duplicated objects. Determining if any object is confusingly similar with the known object further preferably comprises performing one or more of the following process steps: conducting an optical character recognition analysis on the object; conducting a facial analysis on the object; conducting a watermark analysis on the object; conducting a signature analysis on the object; and conducting an object similarity analysis on the object.

Embodiments of the present invention also more specifically provide a method comprising accessing a store that is storing duplicated objects from a database (e.g., an Internet database); and determining if any of the duplicated objects stored in the store are similar with a known object.

Embodiments of the present invention further also more specifically provide a computer-readable storage medium storing program code for causing a processing system to perform the steps of: searching a database for objects; duplicating the objects from the database to produce duplicated objects; storing (e.g., maintaining in memory or transferring into memory) the duplicated objects to produce stored duplicated objects; determining if any stored duplicated object is confusingly similar with a known object.

Embodiments of the present invention also provide for a system for discovering from a database an object which is confusingly similar with a known object comprising: a search engine for searching a database for objects; a duplicator coupled to the search engine for duplicating objects from the database to produce duplicated objects; a store coupled to the duplicator for storing the duplicated objects to produce stored duplicated objects; and determining means, coupled to the store, for determining if any stored duplicated object is confusingly similar with a known object. The system additionally preferably comprises determining the degree of similarity of any stored duplicated object with the known object.

The present invention further also provides a system for discovering from a database an object which is confusingly similar with a known object comprising: means for searching a database for objects; means for duplicating objects from the database to produce duplicated objects; means for storing the duplicated objects to produce stored duplicated objects; and means for determining if any stored duplicated object is confusingly similar with a known object. The

system additionally preferably comprises means for determining the degree of similarity of any stored duplicated object with the known object.

The present inventions also provides a method for determining a degree of similarity between a known object and an object duplicated from a database comprising: duplicating an object from a database to produce a duplicated object; analyzing the content of the duplicated object (e.g., by assigning numbers for each pixel in the duplicated object) to produce a matrix of numbers; producing a model template from a known object; and comparing the model template of the known object with the matrix of numbers to determine the degree of similarity between the duplicated object and the known object. The method for determining a degree of similarity between a known object and an object duplicated from a database preferably additionally comprises one or more of the following process steps: providing a threshold degree of similarity to set a standard for confusingly similarity between the known object and the duplicated object; displaying the degree of similarity if the degree of similarity is at least equal to the threshold degree of similarity; and determining what region of the object the known object is located. The matrix of numbers is created in a RAM when the object (or image) is loaded from storage. The model template is computed and/or created automatically when the first search for object (e.g., a logo) is executed. The model template may be stored in a RAM. Each pixel consists of three numbers representing red, green, and blue. Color depends on algorithms. For example, in object or image searching, the colored image is converted into a grayscale image; subsequently, the actual analysis (or object/image detection) is performed on the grayscale image.

The foregoing provisions along with various ancillary provisions and features which will become apparent to those skilled in the art as the following description proceeds, are attained by the practice of the present invention, a preferred embodiment thereof shown with reference to the accompanying drawings, by way of example only, wherein:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram illustrating a user network access system in accordance with the present invention;

Fig. 2 is a block diagram illustrating details of a company computer system;

Fig. 3A is a schematic diagram of a web crawler coupled to the Internet and its associated servers and further coupled to an object analyzer and storage device;

Fig. 3B is a schematic diagram of a web crawler coupled to the Internet and to an object analyzer and storage device;

Fig. 4 is a block diagram of a RAM device including an operating system, a communication engine, and a browser;

Fig. 5 is a block diagram for an embodiment of the web crawler;

Fig. 6 is a block diagram for another embodiment of the web crawler;

Fig. 7 is a block diagram for an embodiment of the object analyzer and storage device;

Fig. 8 is a block diagram for another embodiment of the object analyzer and storage device;

Fig. 9 is a flowchart in accordance with an embodiment of the invention broadly illustrating a method for sweeping or canvassing a database, such as the Worldwide Internet, for detecting, duplicating, and storing objects (e.g., images, videos, and audio sounds);

Fig. 10 is a flowchart in accordance with an embodiment of the invention broadly illustrating a method for broadly analyzing objects stored after being duplicated from a database, such as the Worldwide Internet;

Fig. 11 is a flowchart in accordance with an embodiment of the invention for illustrating a method for more specifically analyzing the stored objects from Fig. 10;

Fig. 12 is a flowchart in accordance with an embodiment of the invention for illustrating a method for analyzing an image after the stored object has been determined to be an image in accordance with the method schematically illustrated in Fig. 11;

Fig. 13 is a flowchart in accordance with an embodiment of the invention for illustrating a method for analyzing and determining similarity of a known logo with one or more stored logos duplicated from a database, such as the Worldwide Internet;

Fig. 14 is a flowchart in accordance with another embodiment of the invention broadly illustrating a method for online sweeping or canvassing a database for online detecting, analyzing, duplicating, and storing objects;

Fig. 15 is a flowchart broadly illustrating a method for adding and storing URLs which are to be searched in a database;

Fig. 16 is a flowchart in accordance with another embodiment of the invention for illustrating a method for online analyzing and determining similarity of a known logo with any logo detected and analyzed in a database, such as the Worldwide Internet;

Fig. 17 is a pictorial of an image-object for Example I that was duplicated from the Internet and stored in the object storage device; and

Fig. 18 is a pictorial of a known image that was used in Example II to determine if any of the images contained in object storage device were substantially similar to the known image.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The following description is provided to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles, features and teachings disclosed herein.

Referring now to Fig. 1 there is a seen a block diagram illustrating an exemplary user network access system, generally illustrated as **100**, in accordance with various embodiments of the present invention. System **100** includes a company computer system, generally illustrated as **104**, a plurality of servers, generally illustrated as **108**, and an interconnected networks of computers ("Internet") generally illustrated as **112**, for coupling the company computer system **104** to the plurality of servers **108** which include a plurality of web site elements, generally illustrated as **113**. The servers **108** may include any number of servers, such as servers **108a**, **108b**, **108c**, and **108d**. The plurality of web site elements **108** represent web site elements for each server **108a**, **108b**, **108c**, and **108d**. Each server **108a**, etc, and its associated web site elements **108** are typically coupled to a respective computer (not shown) via an internal network signal bus (not shown), and represents a respective possessor or owner of a web page system for advertisement, informational purposes, services, etc., on the Internet **112**. Exemplary advertisement, informational purposes, and services include promotional services, sales information, biographical information, e-mail service programs, address book service programs, calendar service programs, paging service programs, and company database service programs, etc., all of which may include audio sounds, videos, and one or more graphic images (e.g., a reproduction or imitation of a design and text or words including a reproduction or imitation of a person, a thing, a mark, or a symbol) including logos (e.g. non-word elements, a design such as graphic designs, etc), trademarks (e.g., a word, symbol or device pointing distinctly to the origin or ownership of merchandise to which it is applied and legally reserved to the exclusive use of the owner as maker or seller), service marks (e.g. a mark or device used to identify a service offered to customers), faces of people, 2-dimensional objects like animals and cars, etc., all of which may be nonexclusively referred to as "objects." Thus, "objects" comprise images, videos,

audio sounds, and the like. If the user of the company computer system **104** wants to access one of the services of one of the servers **108**, the user applies a known Uniform Resource Locator (URL) to access a web page operated by the possessor of server whose services are to be accessed.

Referring now to Fig. 2 there is seen a block diagram illustrating details of the company computer system **104**. The computer system **104** includes a processor **210** (e.g., a Central Processing Unit) such as a Motorola Power PC® microprocessor or an Intel Pentium® microprocessor. An input device **220**, such as keyboard and mouse, and an output device **230**, such as a Cathode Ray Tube (CRT) display, are coupled via a signal bus **240** to processor **210**. A communications interface **250**, a data storage device **260**, such as Read Only Memory (ROM) or a magnetic disk, and a Random-Access Memory (RAM) **270** are further coupled via signal bus **240** to processor **210**. The communications interface **250** of the computer system **104** is coupled to the Internet **112** as shown in and described with reference to Fig. 1. The computer system **104** also includes an operating system **280**, a web crawler **284**, an object storage device **248**, analyzer parametric Rules **288** for determining similarity, object analyzer and storage device **290**, and a downloading engine **292**.

Referring now to Fig. 3A there is seen a schematic diagram of the web crawler **284** coupled to the Internet **112** (including the servers **108**), and to both the data storage device **260** and the object storage device **248** which latter both in turn are coupled to the object analyzer and storage device **290**. As schematically illustrated in Fig. 3A, the web crawler **284** "walks through" the Internet **112** and sweeps the servers **108**, searching for web objects including images, by automatically following hyperlinks contained in the respective web site elements **113.** It is to be understood that the web crawler **284** may go to any web site, including specified web sites that are not linked (e.g., top level domains (TLD)). The web crawler **284** may also temporarily store URLs, hyperlinks, and copies of objects. An object transfer engine (identified below as **"440"** and **"550"**) may then respectively transfer the web objects and the URLs of the objects to data storage device **260** and to object storage device **248**. Each object contains pixels (e.g. 10,000 or more pixels) and numbers are assigned to each pixel when the object is being analyzed by content. As will be explained below, there are two embodiments for the web crawler **284**. The object analyzer and storage device **290** are coupled to a display or output device **320** and includes the analyzer parametric Rules **288** for determining similarity and the

downloading engine **292** for downloading the web objects and the URLs from the data storage device **260** and the object storage device **248**, respectively. As will be also explained below, there are also two embodiments for the object analyzer and storage device **290** wherein web objects may be analyzed and wherein descriptive information about the content of each web object may be stored. As previously indicated, object analyzer and storage device **290** analyzes web objects by number of pixels in each web object and assigns numbers for each pixel and stores the numbers (i.e., the descriptive information about content of each object). Each pixel consists of three (3) numbers representing the colors red, green, and blue. In Fig. 3B the web crawler **284** is coupled directly to the object analyzer and storage device **290**, instead of being coupled to the object analyzer and storage device **290** via object storage device **248** and data storage device **260**. The web crawler **284** in Fig. 3B is also coupled to a display device **390**. Image or object analysis components employed by the analyzer **290** for each object include, but is not limited to, text (e.g. words and the like), logos, faces (e.g., both human and animal faces, and the like), and two dimensional objects or things (e.g., cars, planes, animals, and the like), and combinations thereof.

The operating system **280** has a program for controlling processing by processor **210**, and may be stored at any suitable location (e.g., in object storage device **260**) and is loaded by the downloading engine **282** into RAM **270** for execution (see Figs. 2 and 4). As best shown in Fig. 4, operating system **280** includes or controls a communication engine **282** for generating and transferring messages including objects to and from the Internet **112** via the communications interface **250**. Operating system **280** further includes or controls an internet engine such as a web browser **246**, e.g., the Netscape™ web browser produced by Netscape, and the Internet Explorer™ web browser produced by the Microsoft Corporation. The web browser **246** may comprise an encryption or decryption engine (not shown in the drawings for encrypting or decrypting messages). The browser **246** further receives web page data including web objects and/or other desired information. The web browser **246** enables a user of the computer system **104** to receive objects including images from the servers **108** via the Internet **112**.

One skilled in the art will recognize that the system **100** may also include additional information, such as network connections, additional memory, additional processors, Local Area Networks (LANs), input/output lines for transferring information across a hardware channel, the Internet **112** or an intranet, etc. One skilled in the art will also recognize that the programs and

data may be received by and stored in the system in alternative ways. For example, a computer-readable storage medium (CRSM) reader such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus **240** for reading a computer-readable storage medium (CRSM) such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system **100** may receive programs and data via a CRSM reader. Further, it will be appreciated that the term "memory" herein is intended to cover all data storage media whether permanent or temporary. Therefore, it will be apparent to those skilled in the art that several variations of the system elements are contemplated as being within the intended scope of the present invention. For example, given processor and computer performance variations and ongoing technological advancements, hardware elements (e.g., multiplexers, etc.) may be embodied in software or in a combination of hardware and software. Similarly, software elements may be embodied in hardware or in a combination of hardware and software. Further, while connection to other computing devices may take place at output device **230** or communications interface **250**, wired, wireless, modem and/or connection or connections to other computing devices (including but not limited to local area networks, wide area networks and the Internet **112**) might be utilized. A further example is that the use of distributed processing, multiple site viewing, information forwarding, collaboration, remote information retrieval and merging, and related capabilities are each contemplated. Various operating systems and data processing systems can also be utilized, however at least a conventional multitasking operating system such as Windows95® or Windows NT® (trademarks of Microsoft, Inc.) running on an IBM® (trademark to International Business Machines, Inc.) compatible computer is preferred and will be presumed for the discussion herein. Input device **220** can comprise any number of devices and/or device types for inputting commands and/or data, including but not limited to a keyboard, mouse, and/or speech recognition.

The web crawler **284** of the present invention sweeps or "walks through" the Internet **112** including servers **108** by automatically following hyperlinks contained in the respective web site elements **113**, or by going to specific web sites that are not linked, such as top level domains (TLD). The web crawler **284** on each web site identifies all web objects and duplicates or copies them from the servers **108** and Internet **112**. Figs. 5 and 6 represent two respective embodiments for the web crawler **284**. Referring to Fig. 5, there is seen one embodiment of the web crawler **284** as including crawler Rules **406** for determining or identifying web objects on the web, an

object search engine **410** for searching all of the servers **108** for web objects in accordance with the crawler Rules **406**, a URL storage device **420** for storing Uniform Resource Locators for each of the web sites, and an object storage device **430** for receiving and temporarily storing web objects that have been identified by the web crawler **284** in accordance with the crawler Rules **406**. The web crawler **246** of Fig. 5 may also include an object transfer engine **440** for transferring the stored web objects from the object storage device **430** to an object data base, such as web object storage device **248**, as well as a URL transfer engine **450** for transferring Uniform Resource Locators from URL storage device **420** to a URL data base, such as data storage device **260**. The web crawler **284** of Fig. 5 continually monitors the entire Internet **112** including the servers **108** for any and all web objects. Thus, this embodiment of the web crawler **284** continually scavenges the Internet **112** including the servers **108** coupled thereto for any and all web objects without making any discernment as to substantial similarity between any object on the Internet **112** and/or servers **108** and any known object.

Referring now to Fig. 6, there is seen another embodiment of the web crawler **284**. This embodiment of the web crawler **284** includes a URL storage device **510** for storing Uniform Resource Locators for each of the web sites, and an object-to-be-searched storage device **520** which receives and stores web objects that are to be searched on the Internet **112** and servers **108** by the web crawler **284**. The user of this embodiment of the web crawler **284** enters or inputs the desired known objects into the object-to-be searched storage device **520** whose substantially similarity is to be searched for on the Internet **112** and servers **108**. This embodiment of the web crawler **284** also includes crawler Rules **526** for determining substantial similarity between the known object(s) stored in the object-to-be searched storage device **520** and any web objects discovered on the Internet **112** and/or servers **108**. The web crawler **284** of Fig. 6 further also includes an object search and comparison engine **530**, an objects-copied-from-web storage device **540**, an object transfer engine **550** and a URL transfer engine **560**. The object search and comparison engine **530** searches in accordance with crawler Rules **526**, the Internet **112** and servers **108** for known objects that are stored in the object-to-be-searched storage device **520**. The engine **530** also compares in accordance with the crawler Rules **526** each web object found on the Internet **112** and/or servers **108** with each known object stored in the object-to-be-searched storage device **520**; and if there is a substantial similarity in accordance with the crawler Rules **526**, the engine **530** downloads (i.e., duplicates or copies) the substantially similar

web object(s) off of the Internet **112** and servers **108** into the objects-copied-from-web storage device **540**. The object transfer engine **550** duplicates and transfers the substantially similar web object(s) from the objects-copied-from-web storage device **540** to a data base, such as object storage device **248.** The URL transfer engine **450** transfers Uniform Resource Locators from URL storage device **510** to a URL data base, such as data storage device **260**. The web crawler **284** of Fig. 6 selectively searches the Internet **112** including the servers **108** for any web objects that are substantially similar to the known object(s) stored in the object-to-be-searched storage device **520**. Thus, for this embodiment of the invention including the web crawler **284**, the web crawler **284** scavenges the Internet **112** and the servers **108** with discernment, looking for any and all web objects that are substantially similar to any and all known objects stored in the object-to-be-searched storage device **520**.

Referring in detail now to Fig. 7 and Fig. 8, there is seen two respective embodiments for the object analyzer and storage device **290**. Referring now to Fig. 7, there is seen one embodiment of the object analyzer and storage device **290** as including the analyzer parametric Rules **288** for determining similarity, an analyzer object comparison engine **730**, the downloading engine **292**, and a descriptive information storage device **710**. The descriptive information storage device **710** contains descriptive information (i.e., mathematical model templates) about one or more known objects for making a determination if the known objects are substantially similar to any of the web objects that were duplicated or copied from the Internet **112** or servers **108** by the web crawler **284**. The analyzer parametric Rules **288** for determining similarity are the rules and parameters that the object analyzer and storage device **290** employs to determine if there is substantial similarity between the descriptive information pertaining to the known objects stored in the descriptive information storage device **710** and the web objects, more specifically the information on the web objects, which is stored in the object storage device **248** after being removed or extracted from the Internet **112** and/or servers **108.** The web-copied or web-duplicated web objects are subsequently either initially stored in object storage device **430**, or in the objects-copied-from-web storage device **540**, or the web-copied web objects by-pass these crawler storage sections and are loaded directly into the object storage device **248**. The downloading engine **297** is capable of downloading web objects (including associated descriptive information on web objects) and URLs from object storage device **248** and data storage device **260**, respectively, into the object analyzer and storage device **290**, more

specifically into the analyzer object storage device **720** of the object analyzer and storage device **290** where the downloaded information is converted into a plurality of numbers from the number of pixels in each web object. As was previously indicated, each pixel in an object is given particular numbers producing a set of numbers which are compared with the mathematical model template of the known object for determining a degree of similarity. Each pixel consists of three (3) numbers representing the colors red, green, and blue. Alternatively, a separate downloading engine (not shown) is employed for downloading URLs from the data storage device **260** into the analyzer object storage device **720** of the object analyzer and storage device **290**.

Once the downloaded web objects and their associated descriptive information arrives in the analyzer object device **720**, the analyzer object comparison engine **730**, under the aegis of the analyzer parametric Rules **288** for determining similarity, makes a comparison between the downloaded web objects (including their associated descriptive information which is in the form of a matrix of numbers from pixels) and the descriptive information (i.e., a template such as a mathematical model template) concerning one or more known objects in the descriptive information storage device **710**. Depending on the degree of substantial similarity, which depends on the analyzer parametric Rules **288**, a match is made between one or more of the downloaded web objects and one or more of the known objects. This information including the corresponding URL(s) for the downloaded web objects may then be provided or displayed through any suitable output device **320**, including a printer or video screen or any of the like. Based on the analyzer parametric Rules **288**, or the analyzer parametric Rules **288** in combination with the analyzer object comparison engine **730**, the degree of similarity may also be provided or displayed. By way of example only, if one or more of the downloaded objects are 90% similar to one or more known objects, such 90% degree of similarity is also provided or displayed. Thus, the analyzer parametric Rules **288**, or the analyzer parametric Rules **288** in combination with the analyzer object comparison engine **730,** enable the degree of similarity between downloaded web objects and known objects to be determined. Typical degree of similarity would be 100%, 95%, 90%, 85%, 80%, or any suitable degree of similarity that is desired. As was previously mentioned, the degree of similarity is preferably determined by comparing a mathematical model template of the known object with a matrix of numbers generated from the number of pixels in each web object, with each pixel consisting of three numbers representing the colors red, green, and blue. The manner in which color in an object or

image is addressed depends on each algorithm. For example, in the logo search exemplified in Fig. 13, a colored image is converted into a grayscale image; then the actual analysis/logo detection is performed on the grayscale image. The matrix of numbers is computed in a RAM, such as RAM **270**, when the object or image is loaded from storage, such as storage device **248**.

Referring now to Fig. 8, there is seen another embodiment of the object analyzer and storage device **290** as including analyzer object comparison engine **830**, and an (optional) analyzer parametric Rules **840**, preferably for "fine tuning" or tweaking any determination of similarity made by the web crawler **284**, more specifically the web crawler **284** of Fig. 6. The analyzer parametric Rules **840** may be optional in the sense that analyzer parametric Rules **840** may not be needed if the crawler Rules **526**, or if the crawler Rules **526** in combination with the object search and comparison engine **530**, are sufficient enough such that the web objects stored in the objects-copied-from-web storage device **540** has the desired degree of similarity with the known objects stored in the object-to-be-searched storage device **520**. In such a case the web-copied or web-duplicated web objects may be displayed through the output or display device **390** (see Fig. 3B), such as any suitable printer and/or video screen or the like. The crawler Rules **526**, or the crawler Rules **526** in combination with the comparison engine **530**, like the analyzer parametric Rules **288** or the analyzer parametric Rules **288** in combination with the analyzer object comparison engine **830**, may also furnish the degree of similarity between web objects in the objects-copied-from-web storage device **540** and the known objects in the objects-to-be-searched storage device **520**. If the crawler Rules **526**, or if the crawler Rules **526** in combination with the object search and comparison engine **530**, are not sufficient for providing a desired degree of similarity (e.g., 100% or 95% degree of similarity), then the analyzer parametric Rules **840**, or the analyzer parametric Rules **840** in combination with the comparison engine **830**, would be employed for "fine tuning" or tweaking the determination of similarity determined by the crawler **284** of Fig. 6, more specifically by the crawler Rules **526**, or by the crawler Rules **526** in combination with the object search and comparison engine **530**, of Fig. 6. Thus, if the degree of similarity detected by the crawler **284** of Fig. 6 is say 50%, then the object analyzer and storage device **290** of Fig. 8, may be used to "fine tune" or tweak this 50% degree of similarity to produce a more sufficient degree of similarity. More specifically, the analyzer parametric Rules **840**, or the analyzer parametric Rules **840** and the comparison engine **830** in combination with the information contained in the descriptive information storage device **710**,

for the object analyzer and storage device **290** of Fig. 8 would be employed to produce a higher degree of similarity (e.g., 90%) between the web object(s) and the known object(s).

The object analyzer and storage device **290** of Fig. 8 may also (optionally) include the downloading engine **292**. If the embodiment of the invention in Fig. 3B is employed such that the web crawler **284** is coupled directly to the object analyzer and storage device **290**, instead of being coupled via object storage device **248** and data storage device **260**, the downloading engine **292** would not be necessary as the object transfer engine **550** of the web crawler could directly transfer any web objects recovered from the Internet **112** and the servers **108** to the analyzer object storage device **820**. The object analyzer and storage device **290** of Fig. 8 also has the descriptive information storage device **710** and an analyzer object storage device **820** which functions comparably to the analyzer object storage device **720**.

Referring now to Fig. 9, there is seen a flowchart for broadly illustrating a method **900** for sweeping or canvassing a database, such as Internet **112**. Storage steps **1000** stores with priorities all URLs whose associated web pages are to be searched by web crawler **284**. Step **910** removes from storage device **1000a** an URL with the highest priority. After removal of the highest priority URL, the web crawler **284** finds the highest priority URL in the Internet **112** and searches for a web page associated with the highest priority URL. If the web crawler **284** in step **920** determines that there is no web page associated with the highest priority URL, then the second highest priority URL is removed from storage device **1000a** and the web crawler **284** repeats the determining step **920** for the second highest priority URL; that is, the web crawler **284** finds the second highest priority URL in the Internet **112** and searches for a web page associated with the second highest priority URL. If the web crawler **284** in step **920** determines that there is no web page associated with the second highest priority URL, the procedure is repeated for a third highest priority URL in storage device **1000a**, and so forth. Alternatively, the web crawler **284** in step **920** determines if there are any more URLs in storage device **1000a** to be searched. In other words, is storage device **1000a** empty of URLs to be searched?

Once it is determined in determining step **920** that a web site or web page **930** is associated with any particular URL, the web page **930** is copied and downloaded by step **940** into web crawler **284**. After downloading by step **940**, all features or elements of the web site or web page **930** are analyzed in analyzing step **950** in accordance with crawler Rules **406** for

determining objects in the downloaded web site or web page **930**. An element of a web site or web page **930** is any hypertext mark-up language (HTML) element by definition. HTML is the standard procedure for writing a web site. Stated alternatively, web crawler **284** analyzes the downloaded web site or web page **930** associated therewith for objects. Step **960** determines if any hyperlinks are discovered for any element; and, if so, the hyperlinks are stored by storage step **1000** (e.g., in storage device **1000a**). Stated alternatively further, the web crawler **284** determines from its associated downloaded web page **930** if any of the elements contained therein include hyperlinks associated therewith; and, if so, the hyperlinks are transferred or downloaded to the storing step **1000** (e.g., downloaded into storage device **1000a**). Hyperlinks effectively execute a "Go To" address wherein the address is the URL associated with the hyperlink. If no hyperlinks are discovered in any particular element by determining step **960**, then determining step **970** determines if the particular element in the downloaded web site or web page **930** includes an object. If one or more objects are found in the particular element being tested, then the object(s) are transferred to object storage device **248**. The URL associated with the object discovered in the particular element is transferred (e.g., is transferred by web crawler **284**) to data storage device **260**. Subsequently, determining step **980** determines if any more elements remain in the downloaded web site or web page **930**. Stated alternatively, the web crawler **284** determines if the last element in the downloaded web page **930** has been tested by determining steps **960** and **970**. If more elements remain, then the next-in-line element is received and determining steps **960** and **970** are performed on the next-in-line element. If the last element of the downloaded web page **930** has been addressed by determining steps **960** and **970**, then the method **900** is repeated for the next highest priority URL from the storage step **1000** (i.e., from storage device **1000a**).

Referring now to Fig. 10, there is seen a flowchart for broadly illustrating a method **1007** for broadly analyzing objects stored after being duplicated from a database, such as the Internet **112**. In step **1020**, the first object to be analyzed for similarity with a known object is removed from the object storage device **248**. After removal, a determination is made by determining step **1030** if all of the necessary metadata (i.e., description information data that describes the object which is preferably a matrix of numbers, with numbers representing a pixel in any stored object) is available for the object. The metadata or a matrix of numbers generated from pixels for any particular stored object is in metadata storage device **1003**. Determining step **1030** (i.e., using a

database query device) searches metadata storage device **1003** for metadata for any particular object. If the necessary metadata for the particular object is not available, then the object is analyzed in step **1001** to develop the necessary metadata. Preferably, object analyzer **1001** develops the necessary metadata by receiving the particular object as input and analyzing that particular object for content (i.e., for metadata content). When the object is a video, each frame of the video will be analyzed for metacontent. Thus, videos are handled as multiple images. After step **1001** and the development of the necessary metadata, a storing step **1040** stores the developed metadata. Preferably the developed metadata is stored in metadata storage device **1003**. Subsequently, the next object is removed by step **1050** from object storage device **248** and the entire procedure is repeated for the next object. If determining step **1030** determines that sufficient metadata exists for any particular object, then steps **1001** and **1040** are bypassed and the next step is step **1050** which is to determine if more objects exist for analyzing. More specifically, a determination is made in step **1050** if object storage device **248** contains more objects which are to be tested to determine if the necessary metadata is available for the particular object. If more objects are available to by analyzed, then step **1060** retrieves the next object from storage device **248** and steps **1030, 1001, 1040** and **1050** are repeated for the next object until determining step **1050** determines that no more objects exist or are available for analysis.

Referring now to Fig. 11 there is seen a flowchart for broadly illustrating the method step **1001** for developing the necessary metadata for any particular object. Object **1109** to be analyzed is input, or otherwise provided, for determining in determining step **1110** if the object **1109** is a video. If object **1109** is not a video then the object **1109** is investigated in determining step **1120** to determine if the object **1109** is an image (e.g., both the texts or words and logos or designs of marks). If the object **1109** is a video then step **1130** analyzes each frame of the video. For each frame in the video, step **1130** employs image analyzer **1002** for analysis and recognition operations on each frame. The results of performing an image analysis and an image recognition operation on each frame of an object video is collected by step **1140** and is transferred in the form of metadata to output step **1150** for storage in step **1040** (see Fig. 10).

The image analyzer **1002** is employed in step **1130** for analyzing each frame of a video after determining step **1110** determines that the object is a video, or the image analyzer **1002** is employed in step **1160** (i.e., the image analyzing step **1160**) after step **1120** determines that the

object itself is an image, e.g., the combination of designs or logos and texts or words in a mark, or the combination of two or more of the following in a mark: texts, logos, facial features, watermarks, signature features, and similarity features. The image analyzer **1002** for embodiments of the present invention performs one or more of the following analyses: OCR (optical character recognition) analysis which recognizes text (e.g., one or more words) in the image; face analysis which detects human or animal faces by employing templates stored in a storage step (identified below as "**1005**"); watermarks analysis which detects and reads embedded watermarks; signature analysis which produces a "digital fingerprint" of the image by calculating one or more numbers, and is employed to identify similar images that have similar "digital fingerprints;" and image similarity analysis which computes one or more numbers that describe the visual similarity of the image to or vis-à-vis images stored in a storing step (identified below as "**1006**"). Each calculated number for signature analysis and for image similarity analysis represents an algorithmic output from a respective algorithm. The more algorithms employed in the signature analysis and in the image similarity analysis, the more algorithmic outputs are produced; and the more algorithmic outputs produced, the more accurate the respective analysis is. The algorithms adjust for size and orientation (e.g., vertical or horizontal) of the object or image. As shown in Fig. 11, the results computed by and/or obtained by the image analyzing step **1160** (e.g., the image analyzer **1002**), along with the results collected by collecting step **1140** of step **1130** are transferred to storing step **1150** were object metadata is stored.

Referring now to Fig. 12, there is seen a flow chart in accordance with an embodiment of the invention for illustrating method **1002** for analyzing an image after step **1120** determines that the object is an image, or for analyzing an image in any frame of a video in accordance with step **1130**. Input step **1210** inputs the image to commence one or more of the following analyzing steps: OCR analyzing step **1220**, face analyzing step **1230**, logo analyzing step **1240**, watermarks analyzing step **1250**, signature analyzing step **1260**, and image similarity analyzing step **1270**. Analyzer parametric Rules **288** are stored (e.g., storage device **288a** stores analyzer parametric Rules **288**). Rules **288** enable the production of image metadata by communicating with and transferring to steps **1220**, **1230**, **1240**, **1250**, **1260**, and **1270** algorithms and/or other parameters which the steps may employ to assist in producing image metadata. OCR analyzing step **1220** receives the pertinent algorithms from analyzer parametric Rules **288** for producing a

plurality of numbers (i.e., OCR algorithmic outputs). For example, one algorithm received from analyzer parametric Rules **288** may be "Caere OCR" which may be purchased commercially from Caere Corporation of Los Gatos, California. As previously indicated, the analyzing steps employ algorithms which adjust for size and orientation of objects or images.

After the OCR analyzing step **1220** has been performed on an image, the face analyzing step **1230** is conducted on the image by receiving the relevant algorithms from analyzer parametric Rules **288** to enable step **1230** is produce the algorithmic outputs (i.e., numbers) for describing any face. For example, one algorithm received from analyzer parametric Rules **288** for analyzing a face may be "Face-It" which may be purchased commercially from Caere Corporation of Los Gatos, California. The more algorithms employed to produce numbers for describing a face, the more accurate the face analysis step **1230** will be. Facial templates (e.g., faces to be searched for on Internet **112**) are stored at storing step **1004** (e.g., in storage device **1004a**). After the face analysis step **1230** has been conducted on an image, the logo analysis step **1240** is conducted on the image. Logo templates (e.g., logos to be searched for on the Internet **112**) are stored at storing step **1005** (e.g., in storage device **1005a**). Logo analysis step **1240** analyses any logos (e.g. design(s) or symbol(s) in a mark) within the image versus the logo templates in storage device **1005a**. A logo template from storage device **1005a** is superimposed over any logo in the image and is similarly produced by template matching.

A watermark analysis may subsequently be conducted on the image by the watermarks analysis step **1250** which receives the relevant algorithms and other parameters from analyzer parametric Rules **288** for detecting and reading embedded watermarks in the image. For example, an algorithm used in the watermarks analysis step **1250** is Digimark Watermarking which is commercially available from Digimark Corporation of Portland Oregon.

After the watermarks analysis step **1250** has been conducted on the image to recognize and analyze the image for watermarks, a signature analysis step **1260** and an image similarity step **1270** is performed on the image. The signature analysis step **1260** receives the pertinent and relevant algorithms from the analyzer parametric Rules **288** and inputs into the algorithms detected variables, such as "color count" and "color distribution" to calculate one or more numbers to produce a "digital fingerprint" which are employed to identify images (i.e., known similar images) that have similar "digital fingerprints." The image similarity analysis step **1270**

receives the pertinent, relevant algorithms for computing one or more numbers (e.g., algorithmic output(s) such as "Color-Histogram-Matching") that describe the visual similarity if any to images in storing step **1006**.

Referring now to Fig. 13, there is seen a flowchart in accordance with an embodiment of the invention for illustrating a method **1300** for analyzing and determining similarity of a known logo **1310** with one or more stored logos which are stored in object storage device **248** after being duplicated from a database, such as the Internet **112**. Step **1320** receives known logo **1310** as input logo-to-search. Stated alternatively, a determination is to be made if known logo **1310** is being used on the Internet **112**; more specifically, if the Internet **112** contains a logo (which could exist in storage device **1005a**) that is confusingly similar to the known logo **1310**. From input step **1320**, logo **1310** is duplicated and stored by step **1330** in storing step **1005** (i.e., logo storage device **1005a**). After duplicating and storing logo **1310** by step **1330**, step **1340** executes method **1007** of Fig. 10 (i.e., the object analyzing process **1007**) to determine if any logos stored in object storage device **248** are confusingly similar to the known logo **1310**. Step **1340** uses the object analyzer **1001** to analyze all objects stored in object storage device **248.** Method step **1340** may be distributed on hundreds of parallel computers. After step **1340** has executed object analysis process **1007**, step **1350** displays the results, along with displaying for the similar logos the corresponding metadata and URL from storage device **1003** and database storage **260** for URLs, respectively.

Referring now to Fig. 14, there is seen a flow chart for broadly illustrating a method **1400** for online sweeping or canvassing a database, such as internet **112**, for online detecting, analyzing, duplicating, and storing objects. For this embodiment of the invention, the web crawler **284** includes its own object analyzer. Storage step **1000** stores with priorities all URLs whose associated web pages are to be searched by web crawler **284**. Step **1410** removes from storage **1000a** an URL with the highest priority. After removal of the highest priority URL, the web crawler **284** finds the highest priority URL in the Internet **112** and searches for a web page associated with the highest priority URL. If the web crawler **284** in step **1420** determines that there is no web page associated with the highest priority URL, then the second highest priority URL is removed from storage **1000a** and the web crawler **284** repeats the determining step **1420** for the second highest priority URL; that is, the web crawler **284** finds the second highest priority URL in the Internet **112** and searches for a web page associated with the second highest

priority URL. If the web crawler **284** in step **1420** determines that there is no web page associated with the second highest priority URL, the procedure is repeated for a third highest priority URL in storage **1000a.**

Once it is determined in determining step **1420** that a web site or web page **1430** is associated with any particular URL, the web page **1430** is copied and downloaded by step **1440** into web crawler **284.** After downloading by step **1440**, all features or elements of the web site or web page **1430** are analyzed in analyzing step **1450** in accordance with crawler Rules **406** for determining objects in the downloaded web site or web page **1430**. As previously indicated, an element of a web site or web page **1430** is a defined HTML element . Stated alternatively, the web crawler **284** includes its own object analyzer for performing analyzes of the downloaded web site or web page **1430** for objects associated therewith. Step **1460** of step **1450** determines if any hyperlinks are discovered for any element, and if so, the hyperlinks are stored by storage step **1000** (e.g., in storage device **1000a**). Stated alternatively, step **1460** of step **1450** of the web crawler **284** determines from associated downloaded web page **930** if any of the elements contained therein include hyperlinks associated therewith; and if so, the hyperlinks are transferred or downloaded to the storing step **1000** (e.g., downloaded into storage device **1000a**). As previously indicated, hyperlinks effectively execute a "Go To" address wherein the address is the URL associated with the hyperlinks. If no hyperlinks are discovered in any particular element by determining step **1460**, then determining step **1470** determines if the particular element in the downloaded web site or web page **1430** includes an object (e.g., an image, an audio, or video). If one or more objects are found in the particular element being tested, then step **1001** (i.e., method **1001** of Fig. 11) is executed for the one or more objects. Step **1480** transfers and/or causes the results to be stored in object metadata storage device **1003**. Subsequently, step **1485** determines if any more URLs exist in storage step **1000** (i.e., storage device **1000a**) having a second highest priority. Stated alternatively, step **1485** tests to determine if storage device **1000a** is empty (i.e., have all URLs been removed for analyzing their associated web pages for objects?). If more URLs exist in storage device **1000a**, then step **1490** retrieves the next highest priority URL from storage device **1000a** and steps **1440** and **1450** are repeated for the next highest priority URL from the storing step **1000** (i.e., from storage device **1000a**).

Referring now to Fig. 15, there is seen a flow chart for broadly illustrating a method **1500** for adding and storing URLs which are to be searched on the Internet **112**. A new URL is

received as input in step **1510**, and then step **1520** transfers and/or otherwise causes the new URL to be stored in the storing step **1000** (i.e., in storage device **1000a**). The first new URL being stored in storage device **1000a** has the highest priority, followed by the second new URL which has the next highest priority, and so forth.

In Fig. 16 there is seen a flow chart in accordance with another embodiment of the invention for illustrating a method **1600** for online analyzing and determining similarity of a known logo with any logo detected in and analyzed from the Internet **112**. For this embodiment of the invention, the web crawler **284** itself possesses the capabilities of doing its own object analysis by having its own object analyzer (i.e., object analyzer **1008**). Step **1620** receives known logo **1610** as input logo-to-search. Stated alternatively, a determination is to be made if known logo **1610** is being used on the Internet **112**; more specifically, if the Internet **112** contains a logo (which could exist in storage device **1005a**) that is confusingly similar to the known logo **1610.** From input step **1620**, logo **1610** is duplicated and stored by step **1630** in storing step **1005** (i.e., logo storage device **1005a**). After duplicating and storing logo **1610** by step **1630**, step **1640** executes method **1008** of Fig. 14 (i.e., web crawler **284** with the object analyzing process **1008**) to determine if any logos on the Internet **112** are confusingly similar to the known logo **1620**. Step **1640** uses the object analyzer **1001** to analyze all objects discovered on the Internet **112** by the web crawler **284**. Method step **1640** may be distributed on hundreds of parallel computers. After step **1640** has executed object analysis process **1008**, step **1560** displays the results, along with displaying for the similar logos the corresponding metadata and URL from storage device **1003** and database storage **260** for URLs, respectively.

The invention will now be illustrated by the following set forth examples which are being given by way of illustration only and not by way of any limitation. All parameters such as, source code, model templates and ID numbers, etc., submitted in these examples are not to be construed to unduly limit the scope of the invention.

Example I

Web crawler **284** was activated to scan the Internet **112** and sweep servers **108**, to search for web objects including images, by automatically following hyperlinks contained in web site elements **113**. The web crawler **284** received an URL from storage device **100a**. The received URL pointed to a web site with the following content which was written in typical HTML language:

```
<html>

<head>
<title>Demonstration</title>

<body>

<p> Demonstration </p>

</div>

<img width=300 height=250 src="./tshirt.jpb"></p>

</div>

<a href="http://www.cobion.com">http://www.cobion.com</a>

</body>

</html>
```

The foregoing web site contained two important elements. The first important element was an image (i.e., both the word(s) and the design(s)/logo(s) in a mark) defined by <img...>. The URL of this image was stored in data storage device **260**. The following information on the image was stored in the object storage device **248**:

a unique image id, for example "970729" (see Fig. 17)
width and height of the image, where width was equal to 300 pixels and height was equal to 250 pixels
current date, for example 12/01/00
image-name, for example "tshirt.jpg"

The second important element in the web site was a hyperlink defined by "<a href=...>." This hyperlink pointed to the web site "http://www.cobion.com" and was stored in storage device **1000**. The stored information associated with this hyperlink was available for use to determine if the stored information including the image was confusingly similar with a known object.

Example II

A search for the "adidas" logo or design (i.e., the known object) was conducted for all objects including images (i.e., both text or words and designs in a mark) contained in the object storage device **248**. The system received the "adidas" logo using the source code (....). For later identification and reference, the system created unique identifier "10001" (see Fig. 18) for the entered "adidas" logo and stored the "adidas" logo in storage device **1005a**. Subsequently, object analysis method **1007** (see Fig. 10) was executed for analyzing the content of objects in object storage device **248.** A matrix of numbers were produced for each object from pixels in each object. Each pixel consists of three (3) numbers representing the colors red, green, and blue. The matrix of numbers were created or computed in RAM **270** when the object(s) were loaded from storage.

The actual analysis for any logo or design in the image of Example I with the "id 970729" took place in object analyzer **1001**. Because the image with "id 970729" was determined to be an image by object analyzer **1001**, image analyzer method **1002** (see Fig. 12) was executed immediately. In method **1002** the logo or design analysis worked in the following manner:

At the beginning the image with "id 970729" was loaded into the RAM **270** (see Fig. 1) of the computer system **104.** RAM **270** created for image with "id 970729" a matrix of numbers comprising:

91	118	109	121	132	145	114	124	110	115	154	187	212	207	205	220	246	249	252	250	25	252	252	253	253	253	244	236	236	242	250	252
106	126	158	157	129	131	184	210	214	226	242	243	250	247	245	247	248	250	250	246	250	246	244	251	252	249	247	241	234	241	248	246
156	203	232	231	230	231	241	242	239	241	248	253	248	246	245	244	245	245	241	242	243	223	219	235	239	232	234	233	222	236	236	227
175	214	233	228	229	231	243	243	240	240	237	233	231	233	234	226	226	216	213	206	147	79	73	167	199	198	197	196	212	195	194	187
167	197	212	221	217	225	224	221	216	216	215	218	213	205	215	207	216	164	111	68	44	26	34	80	167	209	195	186	187	180	169	172
136	169	198	192	200	199	208	216	206	210	211	200	201	196	214	202	204	103	44	40	37	34	34	42	114	170	182	167	177	158	153	175
121	166	187	197	205	195	209	222	211	208	213	193	198	193	207	191	160	131	60	45	23	34	34	36	55	148	178	174	163	155	158	158
121	173	181	216	210	218	201	199	202	195	196	202	202	187	143	84	60	132	117	44	34	34	34	29	29	74	148	170	159	161	161	163
129	182	197	202	214	218	205	203	188	190	199	185	131	89	51	31	26	65	150	89	40	34	34	34	27	27	80	167	151	147	139	158
157	183	193	194	188	199	186	187	181	183	193	122	42	25	34	34	34	40	95	137	62	21	24	34	26	30	43	102	136	143	136	134
151	180	210	193	181	192	189	175	185	170	180	149	57	37	34	34	34	34	49	108	114	40	27	34	34	34	35	57	112	130	131	127
136	180	202	192	183	203	188	192	174	150	115	122	105	42	34	34	34	34	30	60	123	88	19	34	34	34	34	28	71	122	122	121
131	177	200	193	184	181	182	146	97	64	44	55	139	92	39	34	34	34	34	38	79	130	61	35	34	34	18	17	38	88	123	117
172	183	194	189	179	168	84	56	31	34	33	39	83	136	63	24	33	34	34	34	44	113	127	52	36	35	24	17	34	53	114	87
153	183	183	171	178	169	71	38	28	36	34	34	44	92	96	32	25	37	36	34	35	63	107	87	38	36	38	32	36	41	73	76
140	185	162	162	177	171	128	87	78	80	73	75	74	54	62	53	30	83	79	79	81	68	34	96	96	84	85	86	82	78	88	98
152	181	170	160	124	103	119	115	140	172	149	126	127	65	51	80	65	138	134	112	105	87	25	98	156	126	108	111	117	117	132	100
151	199	170	85	30	14	28	19	103	137	76	40	30	26	46	65	45	124	81	41	23	31	17	100	106	50	28	27	33	56	81	39
156	166	112	23	69	106	64	17	83	61	39	80	76	23	32	58	28	86	39	69	93	56	13	85	48	55	85	71	28	51	63	44
148	158	103	13	90	148	105	35	74	45	64	143	134	53	39	61	39	75	39	123	154	92	26	81	30	89	164	129	25	43	90	51
159	158	129	58	42	51	31	5	97	71	23	66	60	25	45	68	52	95	31	38	65	33	14	86	62	24	46	57	13	47	72	57
192	180	172	139	81	48	69	59	118	150	84	32	41	51	62	76	60	137	104	55	30	56	56	107	143	73	31	43	46	63	118	58
233	223	213	216	211	212	222	204	208	223	206	159	166	171	191	204	183	198	204	180	156	197	180	198	196	195	167	160	173	171	180	153
218	220	217	221	224	232	225	221	228	228	226	227	228	218	216	224	225	225	214	229	229	220	214	215	205	212	225	203	202	192	175	183
96	97	95	92	95	109	106	103	105	104	104	106	96	103	96	102	104	92	88	87	86	89	88	82	81	82	87	84	85	81	85	90

Subsequently, the computer system **104** executed the content analysis of the image with "id 970729" by source code:

```
Image:=VC_LoadImage24(path);
```

where "path" was the local location where the image with "id 970729" was stored in object storage device **248** (see Figs. 2 and 3A), which would be "tshirt.jpg" from Example I. "VC_LoadImage" was the function that loaded the image "id 970729" into RAM **270** where a matrix of numbers was produced. After that, image with "id 970729" was compared with all logos contained in the logos-to-search-templates database **1005** (see Fig. 12), including the "adidas" logo with id 10001, by source code:

```
for i := 1 to NumberOfLogos do
begin
 FS_SearchLogo(Image,Logo[i], LogoInfo );
end;
```

The foregoing function "FS_SearchLogo(Image,Logi[i], LogoInfo)" was a computer vision algorithm that searched for the "adidas" logo with id 10001 inside the image with "id 970729" by comparing a mathematical template for the "adidas" logo with id 10001 with the matrix of numbers for image with "id 970729." The computer vision algorithm used for this application was called "Template Matching". The variable "LogoInfo" held the results of the analysis, storing information about the region where the "adidas" logo with id 10001 was found in image with "id 970729", and the similarity of that particular region with the searched "adidas" logo.

The "FS_SearchLogo" function created the mathematical model template automatically when the first search for a logo was executed (on demand). The algorithm used for creating the mathematical model was "Create Template". In this Example II the search was for "adidas" logo with id 10001. The template matching algorithms required a template for the "adidas" logo which was automatically generated from the known "adidas" logo. The Mathematical Model Template for the known "adidas" logo 10001 comprised:

Threshold 423434

0	0	0	0.5	1	0	0
0	0	0.5	2	2	0.5	0
0	0.5	2	1.5	1.5	2	0
0.5	2	1.5	1.5	1.5	2	1
2	1.5	1.5	1.5	1.5	1.5	2
2	2	2	2	2	2	2
0	0	0	0	0	0	0

The following source code caused the discovery of the template for the "adidas" logo 10001 within the matrix of numbers representing the image with "id 970729":

```
// process different image resolutions
 for (_Step=0; _Step<_Steps; _Step++)
 {
     SetCurrentResolution (_Step);

     // try to move template step by step over the whole image from top-left to the
bottom-right position
     for (_RegX=_Left; _RegX<_Right; _RegX++)
     {
         for (_RegY=_Top; _RegY<_Bottom; _RegY++)
         {
                 // calculate match of template
                 double Score = MatchTemplate(10001);

                 // test if the match is above the calculated threshold from the template
                 if (Score > TemplateThreshold)
                       {

                             // store results
                             LogoInfo.Score = Score;
                             LogoInfo.Region = (regX, RegY);

                             return;

                       }
                 }
         }
}
```

The decision whether the template was contained in a particular region of the image with "id 970729" was made by using a "score" (i.e., the degree of similarity) that expressed how good the template matched a region of the image. If the score was above a desired threshold represented by "Threshold 423434" (e.g. 60%), the process assumed the "adidas" logo with id 10001 was found. Thus, embodiments of the present invention provide determining the region of the object (e.g., image with "id 970729") where the known object (e.g., "adidas" logo or design with id 10001) is located.

After the "adidas" logo with id 10001 was successfully found, the results, also called metadata, were stored in object metadata storage device **1003** (see Fig. 10) using the following source code function:

```
function StoreResultsAndGetNextImage(
        const ClientPC: WideString;
        var AnalyseMethod, IDImage:  UINT;
        AnalyseResults: OleVariant;
        tFileLoad, tAnalyse, tCOMCall:  UINT) : WideString;
```

In this Example II, the "adidas" logo with "id 10001" was found in image with "id 970729." The metadata that was stored in object metadata storage device **1003** for this discovery was:

IMAGE_ID	LOGO_ID	REGION	SCORE
970729	10001	Left:166 Top:169 Right:290 Bottom:240	94%

The function for storing the metadata in object metadata device **1003** also retrieved the next image to process from object storage device **248** (see Fig. 10) to begin or repeat the method again.

CONCLUSION

Thus, by the practice of embodiments of the present inventions, there is broadly provided a system and method for deterring and/or detecting Internet abuse of trademarked intellectual property by identifying imposter or look-alike brands, logos/designs, trademarks or service marks, and by identifying unauthorized Internet sales channels. Embodiments of the present inventions also broadly provide speedy data gathering of possible trademark infringement or dilution cases, including providing URL's of suspect sites for tracking or enforcement purposes and showing areas of potential brand erosion in Internet commerce. Embodiments of the present inventions provide a system to search images (e.g., both text or words and designs or logos in marks) in the Worldwide Internet by specifying the visual image content in means of: text contained in any images; logos or designs contained in any images; faces of people contained in any images including face recognition; and two(2) dimensional objects like animals, cars, etc. contained in any images. Embodiments of the present inventions search a database for images which are substantially identical or similar to any known images. Embodiments of the present inventions also enable people to search the Internet for images that have a specified visual content.

While the present invention has been described herein with reference to particular embodiments thereof, a latitude of modification, various changes and substitutions are intended in the foregoing disclosure, and it will be appreciated that in some instances some features of the invention will be employed without a corresponding use of other features without departing from the scope and spirit of the invention as set forth. For example, although the network sites are being described as separate and distinct sites, one skilled in the art will recognize that these sites may be part of an integral site, may each include portions of multiple sites, or may include combinations of single and multiple sites. Furthermore, components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected conventional components and circuits. As previously indicated, connections may be wired, wireless, modem, etc. Therefore, many modifications may be made to adapt a particular situation or material to the teachings of the invention without departing from the essential scope and spirit of the present invention. It is intended that the invention not be limited to the particular embodiment disclosed as the best mode contemplated for carrying out this invention, but that the invention will include all embodiments and equivalents falling within the scope of the appended claims.